

Claims:

What is claimed is:

1. A system for loading software applications, comprising:
a server for executing an software application thereupon, wherein said software application has a number of modules associated therewith;
a control file associated with said software application, wherein said control file specifies a hierarchy of classloaders to be used with said modules;
a deployment mechanism that loads with said software application a selection of said classloaders according to the hierarchy specified by said control file; and,
wherein upon receiving a request to deploy any of said modules the system determines, according to said hierarchy, the minimum number of modules and/or additional modules necessary to deploy or redeploy the software application, and then deploys those modules.
2. The system of claim 1 further comprising a user interface that allows a software developer to specify a subset of said modules to be deployed.
3. The system of claim 1 wherein said modules are any of EJB components, classes, or implementations.
4. The system of claim 2 wherein said user interface allows the software developer to specify a redeploy command that instructs the system that said subset should be redeployed, wherein said redeploy command specifies an software application name and a module associated with the application.
5. The system of claim 4 wherein said redeploy command includes a list of modules relative to the root of the application to be deployed, for redeployment of said modules.

6. The system of claim 5 wherein the module is any of EJB components, class, or implementations.
7. The system of claim 6 wherein the module is a WAR file.
8. The system of claim 6 wherein the module is an EJB impl class.
9. The system of claim 6 wherein the module is an EJB, and wherein each of the classes within that module are redeployed.
10. The system of claim 1 wherein the server provides multiple EJB software applications, each with their own hierarchy of classloaders.
11. A method for loading application components, comprising the steps of:
 - providing an EJB software application thereupon, wherein said EJB software application has a number of modules associated therewith;
 - parsing a control file associated with said EJB software application, wherein said control file specifies a hierarchy of classloaders to be used with said modules;
 - deploying said EJB software application with a selection of said classloaders according to the hierarchy specified by said control file; and,
 - wherein upon receiving a request to deploy any of said modules, determining, according to said hierarchy, the minimum number of modules and/or additional modules necessary to deploy or redeploy the EJB software application, and then deploys those modules.
12. The method of claim 11 further comprising a user interface that allows a software developer to specify a subset of said modules to be deployed.

13. The method of claim 11 wherein said modules are any of EJB components, classes, or implementations.
14. The method of claim 12 wherein said user interface allows the software developer to specify a redeploy command that instructs the method that said subset should be redeployed, wherein said redeploy command specifies an EJB software application name and a module associated with the application.
15. The method of claim 14 wherein said redeploy command includes a list of modules relative to the root of the application to be deployed, for redeployment of said modules.
16. The method of claim 15 wherein the module is any of EJB components, class, or implementations.
17. The method of claim 16 wherein the module is a WAR file.
18. The method of claim 16 wherein the module is an EJB impl class.
19. The method of claim 16 wherein the module is an EJB, and wherein each of the classes within that module are redeployed.
20. The method of claim 11 wherein the server provides multiple EJB software applications, each with their own hierarchy of classloaders.
21. A computer readable medium including instructions stored thereon which when executed cause the computer to perform the steps of
providing an EJB software application thereupon, wherein said EJB software application has a number of modules associated therewith;

parsing a control file associated with said EJB software application, wherein said control file specifies a hierarchy of classloaders to be used with said modules;

deploying said EJB software application a selection of said classloaders according to the hierarchy specified by said control file; and,

wherein upon receiving a request to deploy any of said modules, determining, according to said hierarchy, the minimum number of modules and/or additional modules necessary to deploy or redeploy the EJB software application, and then deploys those modules.

22. The computer readable medium of claim 21 further comprising a user interface that allows a software developer to specify a subset of said modules to be deployed.

23. The computer readable medium of claim 21 wherein said modules are any of EJB components, classes, or implementations.

24. The computer readable medium of claim 22 wherein said user interface allows the software developer to specify a redeploy command that instructs the system that said subset should be redeployed, wherein said redeploy command specifies an EJB software application name and a module associated with the application.

25. The computer readable medium of claim 24 wherein said redeploy command includes a list of modules relative to the root of the application to be deployed, for redeployment of said modules.

26. The computer readable medium of claim 25 wherein the module is any of EJB components, class, or implementations.

27. The computer readable medium of claim 26 wherein the module is a WAR file.

28. The computer readable medium of claim 26 wherein the module is an EJB impl class.
29. The computer readable medium of claim 26 wherein the module is an EJB, and wherein each of the classes within that module are redeployed.
30. The computer readable medium of claim 21 wherein the server provides multiple EJB software applications, each with their own hierarchy of classloaders.